REMARKS

Claims 28-50 are pending.

The claims have been amended to overcome the formal objections raised by the Examiner. Applicants' attorneys thank the Examiner for his courtesy in noting the needed corrections.

Claims 28-30, 38-41 and 50, which include each of the independent Claims 28, 39 and 50, are rejected as being anticipated by Buch, et al., U.S. 7,114,011. Claims 31-37 and 42-48 are rejected as unpatentable over Buch.

Claims 32-37 and 43-48 are directed to the processing queue for the data. Independent claim 28 has been amended to incorporate the subject matter of claim 32, which has been cancelled, and independent claim 39 amended to incorporate the processing queue subject matter of claim 43, which has been cancelled. Claim 50 also is amended to recite the queue for each CPU. Other claims of these groups recite various degrees of detail of the serial operation of the queue. Reference is made to [0041-0042] and [0054] which describes the operation of the queue.

The independent claims 28, 39 and 50 each also recite that there are a plurality of interfaces and that each one of a plurality of interfaces is bound to only one of the plurality of CPUs and that each CPU has its own processing queue. A processing queue has a kernel data structure, e.g., a serialization queue type and a worker thread that is controlled by the queue. Thus, a CPU is not arbitrarily interrupted by incoming data (e.g., a packet) received via the network interface. This prevents contention for individual locks and queues being formed at each protocol layer [0014]. The claims further recite that a separate instance of the application is run on each of the plurality of interfaces.

Thus, there is a vertical perimeter of a CPU having its own processing queue and at its interface on which a separate instance of the application is run. This creates a vertical path that includes hardware and software elements plus any synchronization/ serialization mechanisms. It should be understood that the typical protocol implementation (TCP/IP) states that needs to be protected from multiple thread access and making sure that protection mechanism (s-queue in the application) is also part of the vertical path. This provides a significant advantage.

Buch is directed to a streaming data server. The streaming broadcast data delivered from a source is to be distributed to a plurality of clients connected to the network via various interface cards as demanded by the various clients. Buch neither teaches nor suggests the same type of vertical perimeter arrangement as is set forth in the claims of the application, i.e., an interface designated to a CPU which runs its own instance of the application.

Buch has a multi-processor scalable streaming data server. That is, Buch adds CPUs horizontally to be able to service more clients relative to a single data source, the data stream. The Examiner, referring to column 1, lines 14-22 of Buch, considers that providing streaming data over the internet corresponds to a method of providing services of an application by way of an instance of an application for each of Buch's CPUs. As explained below, this does not correspond to the language of the amended independent claims 28, 39 and 50 of the application that set forth that there is a separate instance ([0012]) of the application for each one of the interfaces and each of the plurality of interfaces is designated to only one of the CPUs that has its own processing queue. The memory caches of Buch are not the queues of the invention. The disadvantages of caches is described at [0010] of the application.

In Buch, Fig. 4, referring to column 4, lines 42-65 and using the same step numbers, in step 1 the incoming streaming data received at NIC0 is sent to the IN_BUF of a memory 430 of a chip set 420. In step 2, the data is read out from the IN_BUF server by the server thread A of CPU0. In step 3 the server thread A of CPU0 then copies the data read from the IN_BUF of memory 430 to the memory's OUT_BUF. In step 4, the NIC 3 then generates a hardware interrupt which is serviced by the CPU 3 and the DPC (delayed process call) is registered. It is not stated in the Buch application as to what initiates the call to NIC 3. In step 5, a DPC handler executes on CPU 3 and there is binding of the interrupt and DPC. In step 6 the DPC handler invokes server thread B which runs on CPU 3, that there is server thread B/buffer affinity. In step 7, the server thread B completes packet assembly and locks the memory 430 OUT_BUF ready for the NIC 3 to read out the data, the OUT_BUF and stream it to the clients connected to NIC 3.

In a data processing system, the TCP/IP (protocol) state of the listener and active connections is protected by means of different synchronization mechanisms like locks, serialization queues, etc. In multi processor system, to achieve scalability, these synchronization mechanism are

typically per CPU basis and a listener for a particular IP address is protected by the serialization (processing), or squeue, on the CPU its executing on. For example, say a application instance dealing with IP address A is running on CPU 1. Thus, all connection and protocol state is protected by the serialization (processing) queue on CPU 1. But if the IP address A is tied of NIC 3 which is in turn tied to CPU 3, then a problem occurs. All incoming data packets are coming on CPU 3 but they need to be processed on CPU 1 since the application and protocol state is protected by the squeue of CPU 1.

The present invention overcomes this problem by providing a true vertical perimeter. In accordance with the invention, when an application starts a listener for IP address A which is tied to NIC 3 and CPU 3, that application instance is actually moved to CPU 3 itself and it is tied to the squeue on CPU 3 itself, thus making sure that all inbound and outbound data goes through CPU 3, squeue on CPU 3, NIC 3.

Applicant stresses that other than lining up the hardware elements together (NIC and CPUs), the kernel queue and threads together are lined up together as well (along with application instance) to really achieve a vertical path through hardware and the software. This achieves real scalability in practice on multi CPU systems because of lower lock contention, etc.

The Examiner considers that a request made by a client to extract data from the stream corresponds to the claimed subject matter of a separate instance (copy) of an application bound to a CPU. In applicants' invention, the "instance" is the application itself. In Buch one client could be requesting one type of data A from the stream via one interface and another client requesting different data B from the stream via another interface. A request for data, which can be different, is not an application. In Buch, each interface is not bound to a separate instance.

What is disclosed in Buch is not the vertical perimeter processing set forth in the claims which each defines a stack of an interface, CPU and application instance that is able to accomplish end to end processing of data. In Buch there does not appear to be a separate instance of the application bound to each of the plurality of interfaces. Also, the processing of the incoming data stream is done at least in part on a horizontal level since the data supplied to each of the CPUs passes through the common memory BUF_IN and BUF_OUT of memory 430. That is, Buch would

appear to have to perform each of steps 1-7 described above for each call made by a client from a different node.

Further, Buch does not teach that his NICs (network interface cards) each has its own separate network address, as set forth in claims 29-31 and 40-42. This permits an incoming data (packet) to be processed from beginning to end by a single processor that has been assigned to the network interface that receives the data [0014]. Therefore, a CPU is designated (bound) to a particular address. This permits an application instance to be a listener [0037] that listens for the network addresses assigned to the network interface cards [0030]. None of these advantageous features can be achieved by Buch.

In summary, the independent claims 28, 39 and 50 each defines an invention that has advantages over what is taught by Buch. Therefore, these claims are patentable and should be allowed. The dependent claims of the group rejected as being anticipated by Buch define further features to the novelty of the invention. Therefore they also should be allowed.

All of the claims in the application are patentable and should be allowed. Therefore, the application should be passed to issue.

CONCLUSION

Applicant respectfully submits that the application is in condition for allowance. Favorable consideration on the merits and prompt allowance are respectfully requested. In the event any questions arise regarding this communication or the application in general, the Examiner is invited to contact Applicant's undersigned representative at the telephone number listed below.

Dated: May 22, 2008

Respectfully submitted,

By _____
John W. Branch
Registration No.:
DARBY & DARBY P.C.
P.O. Box 770
Church Street Station
New York, New York 10008-0770
Phone (206) 262-8900 • Fax (212) 527-7701
Attorneys/Agents For Applicant